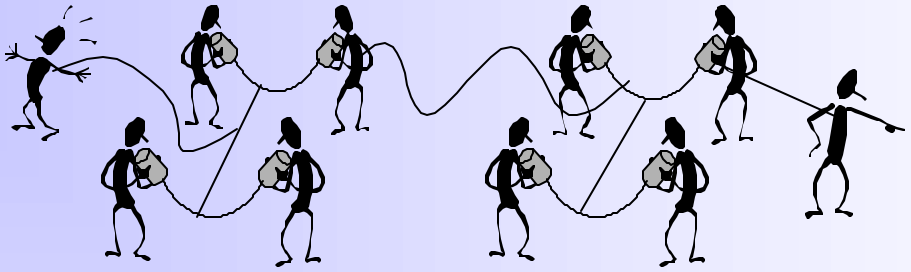
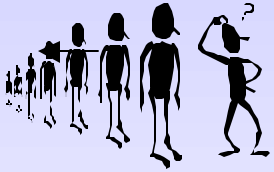
CLIQUES: Security for Dynamic Peer Groups

Gene Tsudik

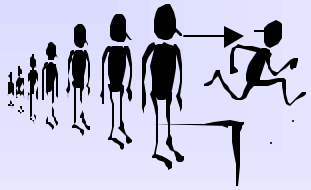
Yongdae Kim



Formation



Member add



Member leave

Group fusion

Giuseppe Ateniese

Group fission

Damian Hasse

Background

Targeted environment

- Relatively small groups
- Dynamic membership
- No hierarchy
- Many-to-Many
- Collaborative applications

Problem:

how to obtain security in peer groups with **dynamic** membership and **decentralized** control?

Security Services Provided

- Decentralized authenticated group key agreement with provable security based on group Diffie-Helman: each member contributes equally to group key
- Membership changes: single member, many members and sub-groups
- Membership authentication: based on knowledge of key-share
- Authenticated join/leave: requires long-term DH credentials

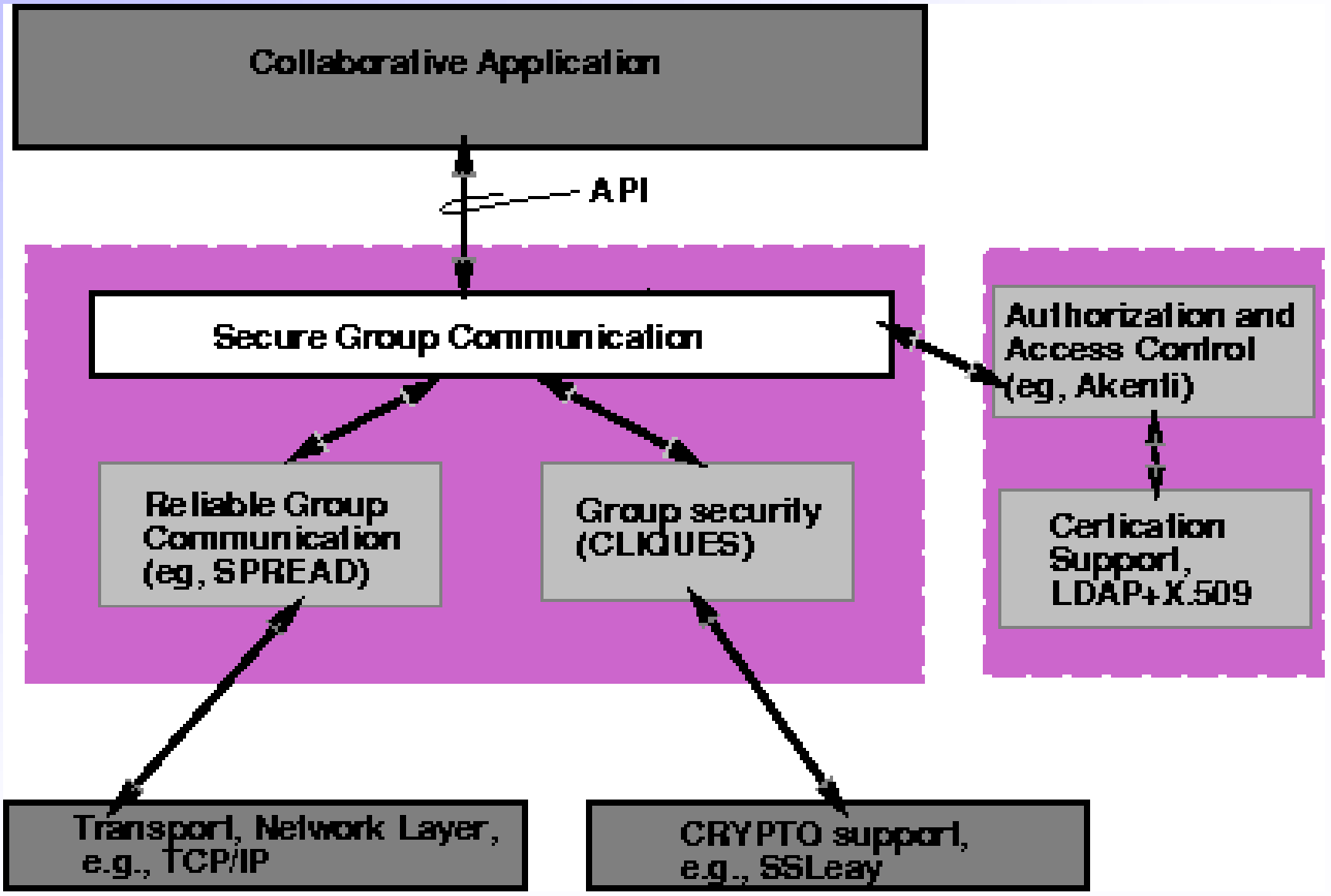
Other pieces of the puzzle

- Certification infrastructure
- Reliable group communication subsystem
- Membership Authorization / Access control

STATUS

- Initial Key Agreement [stw96]
- Auxiliary Key Agreement (membership changes) [stw98]
- Authenticated Key Agreement [ast98]
- JAVA implementation (rel. 0.0)
- C implementation (rel. 0.1) coupled with JHU's SPREAD
- CLQ_API: coding completed (rel. 1.0) mid-Feb
- Testing and integrating with SPREAD [tbd99]
- Current performance results (0.024sec/exp!!! Yuck...)
- Integration with TOTEM on-going (LBL)
- Integration with AKENTI: near future

Proposed Architecture



CLQ_API prerequisites

Underlying group communication subsystem must provide reliable synchronized event notification for:

- group joins
- group leaves
- partitions
- node failures or disconnects
- merges (partition heals)

CLQ_API

called by a new group member who received a NEW_MEMBER message from the current controller.

clq_join (ctx, member_name, group_name, input, output);

called by the current controller to hand over group context to a new member (who becomes next controller).

clq_pass_ctx (ctx, member_name, output);

called by every member upon reception of a KEY_UPDATE_MESSAGE from the current group controller

clq_update_ctx (ctx, input);

CLQ_API (contd)

called by every group member after member leaves or partition occurs; removes all valid members in member_list from the group_member_list. Only controller gets output token

clq_leave (ctx, member_list[], output);

called only by controller when group_secret needs to be updated.

clq_refresh_key (ctx, output);

Almost done...

Lessons learned, challenges and directions:

- Paper protocols <> real protocols
- Incremental formation of groups
- Security, group comm not a simple composition
- Difficulty of handling many merging sub-groups
- Group size limits (100?)
  - other DH-like keys
  - elliptic curve duals

- Two-tiered groups (few senders, many receivers)
- Group membership policy (Auth + AC)
- Group certification: group key, membership, etc.
  - Dynamic membership?
  - Individual vs opaque certificates?

Last chart...

Tech Transfer:

- IBM
- JHU
- LBNL (DoE)
- IRTF->IETF?

CLIQUES web page: http://www.isi.edu/div7/cliques

- API documentation
- Publications
- Presentations

API code by request from gts@isi.edu

Collaborators:

- IBM Research, Johns Hopkins, LBNL, Nortel

Secure Spread SD